

```

#define gen_max 1000
#define D 7
#define NP 30
#define F 0.9
#define CR 0.6          /* 0.7 */
#define inibound_l 0.0
#define inibound_h 1.0

/*----Constant for rnd_uni()-----*/
#define IM1 2147483563
#define IM2 2147483399
#define AM (1.0/IM1)
#define IMM1 (IM1-1)
#define IA1 40014
#define IA2 40692
#define IQ1 53668
#define IQ2 52774
#define IR1 12211
#define IR2 3791
#define NTAB 32
#define NDIV (1+IMM1/NTAB)
#define EPS1 1.2e-7
#define RNMIX (1.0-EPS1)

#include<stdlib.h>
#include<stdio.h>
#include<time.h>
#include<math.h>
#include<conio.h>

double evaluate(double [],long *);
double evaluate(double tmp[],long *nfe)
{
    double cost; (*nfe)++;          /* tmp[0]=x1, tmp[1]=x2, tmp[2]=x3 */

    cost=pow((tmp[0]-1), 2)+pow((tmp[1]-2), 2)+pow((tmp[2]-3), 2)+pow(((tmp[3])-1), 2)+pow(((tmp[4]
]-1), 2)+pow(((tmp[5])-1), 2)-log((tmp[6]+1));
    return(cost);

} /***** end of evaluate() *****/

float rnd_uni(long *);
float rnd_uni(long *idum)
{
    long j; long k;
    static long idum2=123456789;
    static long iy=0;static long iv[NTAB]; float temp;
    if(*idum<=0)
    {
        if(-(*idum)<1) *idum=1; else *idum=-(*idum); idum2=(*idum);
        for(j=NTAB+7;j>=0;j--)
        {
            k=(*idum)/IQ1;
            *idum=IA1*(*idum-k*IQ1)-k*IR1;
            if(*idum<0) *idum+=IM1;
            if(j<NTAB) iv[j]=*idum;
        }
        iy=iv[0];
    }
    k=(*idum)/IQ1;
    *idum=IA1*(*idum-k*IQ1)-k*IR1;
    if(*idum<0) *idum+=IM1;
    k=idum2/IQ2;
    idum2=IA2*(idum2-k*IQ2)-k*IR2;
    if(idum2<0) idum2+=IM2;
    j=iy/NDIV; iy=iv[j]-idum2; iv[j]=*idum;
    if(iy<1) iy+=IMM1;
    if((temp=AM*iy)>RNMIX) return RNMIX;
    else return temp;
}

void main()
{
    int i,j,k,a,b,c,good,count=0,seed; long nfe=0;
    double x1[50][10],x2[50][10],cost[50],trial[10],cost_trial,pen;
    double lhs2,lhs3,lhs4,lhs5,lhs6,lhs7,lhs8,lhs9,costmin,lhs1,costmax;

```

```

    clock_t start, end;
    printf("\nseed=");
    scanf("%d",&seed);
    long rnd_uni_init= -(long)seed;    start = clock();

for (i=0;i<NP;i++)
{
    for (j=0;j<D;j++)
    {
        x1[i][j]=inibound_l + rnd_uni(&rnd_uni_init)*(inibound_h-inibound_l);
    }
    if(x1[i][3]>=0.5)    x1[i][3]=1.0;    else x1[i][3]=0.0;
    if(x1[i][4]>=0.5)    x1[i][4]=1.0;    else x1[i][4]=0.0;
    if(x1[i][5]>=0.5)    x1[i][5]=1.0;    else x1[i][5]=0.0;
    if(x1[i][6]>=0.5)    x1[i][6]=1.0;    else x1[i][6]=0.0;
    pen=0.0;

    lhs1=x1[i][0]+x1[i][1]+x1[i][2]+(x1[i][3])+(x1[i][4])+(x1[i][5]);
    if(lhs1>5.0)
    {
        pen=lhs1*10.0;
        cost[i]=evaluate(x1[i], &nfe);
        cost[i]=cost[i]+pen;
        continue;
    }

    lhs2=pow(x1[i][0], 2)+pow(x1[i][1], 2)+pow(x1[i][2], 2)+pow((x1[i][5]), 2);
    if(lhs2>5.5)
    {
        pen=lhs2*10.0;
        cost[i]=evaluate(x1[i], &nfe);
        cost[i]=cost[i]+pen;
        continue;
    }
    lhs3=x1[i][0]+(x1[i][3]);
    if(lhs3>1.2)
    {
        pen=lhs3*10.0;
        cost[i]=evaluate(x1[i], &nfe);
        cost[i]=cost[i]+pen;
        continue;
    }
    lhs4=x1[i][1]+(x1[i][4]);
    if(lhs4>1.8)
    {
        pen=lhs4*10.0;
        cost[i]=evaluate(x1[i], &nfe);
        cost[i]=cost[i]+pen;
        continue;
    }
    lhs5=x1[i][2]+(x1[i][5]);
    if(lhs5>2.5)
    {
        pen=lhs5*10.0;
        cost[i]=evaluate(x1[i], &nfe);
        cost[i]=cost[i]+pen;
        continue;
    }
    lhs6=x1[i][0]+(x1[i][6]);
    if(lhs6>1.2)
    {
        pen=lhs6*10.0;
        cost[i]=evaluate(x1[i], &nfe);
        cost[i]=cost[i]+pen;
        continue;
    }
    lhs7=pow(x1[i][1], 2)+pow((x1[i][4]), 2);
    if(lhs7>1.64)
    {
        pen=lhs7*10.0;
        cost[i]=evaluate(x1[i], &nfe);
        cost[i]=cost[i]+pen;
        continue;
    }
}

```

```

lhs8=pow(x1[i][2], 2)+pow((x1[i][5]), 2);
if(lhs8>4.25)
{
pen=lhs8*10.0;
cost[i]=evaluate(x1[i], &nfe);
cost[i]=cost[i]+pen;
continue;
}
lhs9=pow(x1[i][2], 2)+pow((x1[i][4]), 2);
if(lhs9>4.64)
{
pen=lhs9*10.0;
cost[i]=evaluate(x1[i], &nfe);
cost[i]=cost[i]+pen;
continue;
}

if(lhs1<=5.0 && lhs2<=5.5 && lhs3<=1.2 && lhs4<=1.8 && lhs5<=2.5 && lhs6<=1.2 && lhs7<=1.64 && lhs8<=4.25 && lhs9<=4.64)
cost[i]=evaluate(x1[i], &nfe);
}

costmin=cost[0];
for(i=1;i<NP;i++)
{ if(costmin>cost[i])
costmin=cost[i];
}

while (count<gen_max)
{
for (i=0;i<NP;i++)
{
do a=int ((rnd_uni(&rnd_uni_init))*NP); while (a==i);

do b=int (rnd_uni(&rnd_uni_init)*NP); while (b==i || b==a);

do c=int (rnd_uni(&rnd_uni_init)*NP); while (c==i || c==a || c==b);

j=int (rnd_uni(&rnd_uni_init)*D);

for (k=1;k<=D;k++)
{
if(rnd_uni(&rnd_uni_init)<CR || k==D)
{
trial[j]=x1[c][j]+F*(x1[a][j]-x1[b][j]);
}
else trial[j]=x1[i][j];

if(trial[3]>=0.5) trial[3]=1.0; else trial[3]=0.0;
if(trial[4]>=0.5) trial[4]=1.0; else trial[4]=0.0;
if(trial[5]>=0.5) trial[5]=1.0; else trial[5]=0.0;
if(trial[6]>=0.5) trial[6]=1.0; else trial[6]=0.0;

if(trial[j]<0.0) trial[j]=0.0;

j=(j+1)%D;
}
}

pen=0.0;

lhs1=trial[0]+trial[1]+trial[2]+(trial[3])+(trial[4])+(trial[5]);
if(lhs1>5.0)
{
pen=lhs1*10.0;
cost_trial=evaluate(trial, &nfe);
cost_trial=cost_trial+pen;
if(cost_trial<=cost[i])
{
for (j=0;j<D;j++)
x2[i][j]=trial[j];
cost[i]=cost_trial;
if(cost_trial<costmin)
{
costmin=cost_trial;
/* imin=i;
}
}
}
}

```

```

        assignd(best,trial); */
    }
}
else for (j=0;j<D;j++)
    x2[i][j]=x1[i][j];
continue;
}
lhs2=pow(trial[0], 2)+pow(trial[1], 2)+pow(trial[2], 2)+pow((trial[5]), 2);
if(lhs2>5.5)
{
pen=lhs2*10.0;
cost_trial=evaluate(trial, &nfe);
cost_trial=cost_trial+pen;
if(cost_trial<=cost[i])
{
for (j=0;j<D;j++)
x2[i][j]=trial[j];
cost[i]=cost_trial;
if(cost_trial<costmin)
{
costmin=cost_trial;
/* imin=i;
assignd(best,trial); */
}
}
else for (j=0;j<D;j++)
x2[i][j]=x1[i][j];
continue;
}
lhs3=trial[0]+(trial[3]);
if(lhs3>1.2)
{
pen=lhs3*10.0;
cost_trial=evaluate(trial, &nfe);
cost_trial=cost_trial+pen;
if(cost_trial<=cost[i])
{
for (j=0;j<D;j++)
x2[i][j]=trial[j];
cost[i]=cost_trial;
if(cost_trial<costmin)
{
costmin=cost_trial;
/* imin=i;
assignd(best,trial); */
}
}
else for (j=0;j<D;j++)
x2[i][j]=x1[i][j];
continue;
}
lhs4=trial[1]+(trial[4]);
if(lhs4>1.8)
{
pen=lhs4*10.0;
cost_trial=evaluate(trial, &nfe);
cost_trial=cost_trial+pen;
if(cost_trial<=cost[i])
{
for (j=0;j<D;j++)
x2[i][j]=trial[j];
cost[i]=cost_trial;
if(cost_trial<costmin)
{
costmin=cost_trial;
/* imin=i;
assignd(best,trial); */
}
}
else for (j=0;j<D;j++)
x2[i][j]=x1[i][j];
continue;
}
lhs5=trial[2]+(trial[5]);
if(lhs5>2.5)
{

```

```

pen=lhs5*10.0;
cost_trial=evaluate(trial, &nfe);
cost_trial=cost_trial+pen;
if(cost_trial<=cost[i])
{
    for (j=0;j<D;j++)
        x2[i][j]=trial[j];
    cost[i]=cost_trial;
    if(cost_trial<costmin)
    {
        costmin=cost_trial;
        /* imin=i;
        assignd(best,trial); */
    }
}
else for (j=0;j<D;j++)
    x2[i][j]=x1[i][j];
continue;
}
lhs6=trial[0]+(trial[6]);
if(lhs6>1.2)
{
    pen=lhs6*10.0;
    cost_trial=evaluate(trial, &nfe);
    cost_trial=cost_trial+pen;
    if(cost_trial<=cost[i])
    {
        for (j=0;j<D;j++)
            x2[i][j]=trial[j];
        cost[i]=cost_trial;
        if(cost_trial<costmin)
        {
            costmin=cost_trial;
            /* imin=i;
            assignd(best,trial); */
        }
    }
    else for (j=0;j<D;j++)
        x2[i][j]=x1[i][j];
    continue;
}
lhs7=pow(trial[1], 2)+pow((trial[4]), 2);
if(lhs7>1.64)
{
    pen=lhs7*10.0;
    cost_trial=evaluate(trial, &nfe);
    cost_trial=cost_trial+pen;
    if(cost_trial<=cost[i])
    {
        for (j=0;j<D;j++)
            x2[i][j]=trial[j];
        cost[i]=cost_trial;
        if(cost_trial<costmin)
        {
            costmin=cost_trial;
            /* imin=i;
            assignd(best,trial); */
        }
    }
    else for (j=0;j<D;j++)
        x2[i][j]=x1[i][j];
    continue;
}
lhs8=pow(trial[2], 2)+pow((trial[5]), 2);
if(lhs8>4.25)
{
    pen=lhs8*10.0;
    cost_trial=evaluate(trial, &nfe);
    cost_trial=cost_trial+pen;
    if(cost_trial<=cost[i])
    {
        for (j=0;j<D;j++)
            x2[i][j]=trial[j];
        cost[i]=cost_trial;
        if(cost_trial<costmin)
        {

```

```

        costmin=cost_trial;
        /* imin=i;
        assignd(best,trial); */
    }
} else for (j=0;j<D;j++)
    x2[i][j]=x1[i][j];
continue;
}
lhs9=pow(trial[2], 2)+pow((trial[4]), 2);
if(lhs9>4.64)
{
pen=lhs9*10.0;
cost_trial=evaluate(trial, &nfe);
cost_trial=cost_trial+pen;
if(cost_trial<=cost[i])
{
for (j=0;j<D;j++)
x2[i][j]=trial[j];
cost[i]=cost_trial;
if(cost_trial<costmin)
{
costmin=cost_trial;
/* imin=i;
assignd(best,trial); */
}
}
else for (j=0;j<D;j++)
x2[i][j]=x1[i][j];
continue;
}

if(lhs1<=5.0 && lhs2<=5.5 && lhs3<=1.2 && lhs4<=1.8 && lhs5<=2.5 && lhs6<=1.2 && lhs7<=1.64 && lhs8<=4.25 && lhs9<=4.64)
{ cost_trial=evaluate(trial, &nfe);
if(cost_trial<=cost[i])
{
for (j=0;j<D;j++)
x2[i][j]=trial[j];
cost[i]=cost_trial;
if(cost_trial<costmin)
{
costmin=cost_trial;
/* imin=i;
assignd(best,trial); */
}
}
else for (j=0;j<D;j++)
x2[i][j]=x1[i][j];
continue;
}

} /***** end of for loop *****/

for (i=0;i<NP;i++)
{
for (j=0;j<D;j++)
x1[i][j]=x2[i][j];
}

costmax=cost[0];
for(i=1;i<NP;i++)
{ if(costmax<cost[i])
costmax=cost[i];
}
costmin=cost[0];
for(i=1;i<NP;i++)
{ if(costmin>cost[i])
costmin=cost[i];
}

if((costmax-costmin)<0.00001)
break;
count++;

} /***** end of while loop *****/

```

```
        end = clock();
for(i=0;i<NP;i++)
{
    for(j=0;j<D;j++)
        printf("u[%d]=%lf",j,(x1[i][j]));
        printf("cost[%d]=%lf",i,cost[i]);
    }
printf("NFE=%ld\n",nfe);
printf("The time was: %f\n", (end - start) / CLK_TCK);
printf("costmax is=%lf",costmax);
} /***** end of main() *****/
```